

## CLAIMS

What is claimed is:

1 A method for automatically instantiating built-in-system test (BIST) modules in memory designs, comprising the steps of:

- (a) providing a server over a network that integrates a set of design tools, including an automated front-end software process and an automated back-end software process;
- (b) allowing a user to access the server over the network and enter a request for a memory design;
- (c) executing the front-end software process to automatically generate a netlist of a BIST from the user request; and
- (d) executing the back-end software process to automatically generate a placement and route view of the BIST.

2 The method of claim 1 wherein step (b) further includes the step of allowing the user to access the server via a standard Web browser.

3 The method of claim 2 wherein step (b) further includes the step of allowing a user to enter the request by entering data into a form and identifying an input script file containing a series of user input command lines.

4 The method of claim 3 wherein step (b) further includes the step of receiving

from the request a number of words and bits in the memory.

5 The method of claim 4 wherein step (c) further includes the step of in putting the command lines from the input script file into appropriate software design tools.

6 The method of claim 1 wherein step (d) further includes the step of: using an iterative algorithm that performs the steps of:

- (i) generating an initial size estimate of an area needed for the memory;
- (ii) allocating a memory having an area of that size;
- (iii) performing placement and routing; and
- (iv) assessing whether the allocated area is sufficient, and if not, incrementing the size of the memory and iterating again.

7 The method of claim 6 further including the step of providing a random access memory (RAM) as the memory, and generating a RAMBIST for the memory.

8 A computer implemented method for performing automatic placement and routing by a back-end software process, the method comprising the steps of:

- (a) generating an initial size estimate of an area needed for the memory;
- (b) allocating a memory of the estimated size;

- (c) performing placement and routing; and
- (d) assessing whether the allocated memory is sufficient, and if not, incrementing the size of the memory and iterating again. 112(2) *structure*

5           9 The method of claim 8 wherein step (a) further includes the step of receiving a netlist comprising a logical description of cells and interconnects in a design as input.

10 The method of claim 9 wherein step (a) further includes the step of generating the initial size estimate by counting a number of cells in the netlist and dividing by constant.

11 The method of claim 10 wherein step (c) further includes the step of:

- (i) performing an initial placement of cells in the allocated memory;
- (ii) determining if the initial placement of cells in the allocated memory is successful;
- (iii) if the initial placement of cells in the allocated memory is successful, performing global routing for large signals;
- (iv) determining if the global routing in allocated memory is successful; and
- (v) if the global routing and allocated memory is successful, performing detail routing for local nets.

12 The method of claim 11 further including the step of using a heuristic algorithm to determine success and failure.

5 13 The method of claim 12 further including the step of incrementing the allocated memory by a minimum amount when failure is determined.

14 The method of claim 13 further including the step of incrementing the allocated memory by one row at a time until it is determined that the allocated memory can be placed and routed without size or timing violations.

15 A computer readable medium containing program instructions for <sup>112(2)</sup>  
automatically a method for automatically instantiating built-in-system test (BIST)  
modules in memory designs, comprising the steps of:

- (a) providing a server over a network that integrates a set of design tools, including an automated front-end software process and an automated back-end software process;
- (b) allowing a user to access the server over the network and enter a request for a memory design;
- (c) executing the front-end software process to automatically generate a netlist of a BIST from the user request; and
- (d) executing the back-end software process to automatically generate a placement and route view of the BIST.

16 The computer readable medium of claim 15 wherein instruction (b) further includes the instruction of allowing the user to access the server via a standard Web browser.

5

17 The computer readable medium of claim 16 wherein instruction (b) further includes the instruction of allowing a user to enter the request by entering data into a form and identifying an input script file containing a series of user input command lines.

18 The computer readable medium of claim 17 wherein instruction (b) further includes the instruction of receiving from the request a number of words and bits in the memory.

19 The computer readable medium of claim 18 wherein instruction (c) further includes the instruction of in putting the command lines from the input script file into appropriate software design tools.

20 The computer readable medium of claim 15 wherein instruction (d) further includes the instruction of: using an iterative algorithm that performs the instructions of:

- (i) generating an initial size estimate of an area needed for the memory;

- (ii) allocating a memory having an area of that size;
- (iii) performing placement and routing; and
- (iv) assessing whether the allocated area is sufficient, and if not, incrementing the size of the memory and iterating again. 112(2)

5

21 The computer readable medium of claim 20 further including the instruction of providing a random access memory (RAM) as the memory, and generating a RAMBIST for the memory.

22 A computer readable medium containing program instructions for performing automatic placement and routing by a back-end software process, the method comprising the steps of:

- (a) generating an initial size estimate of an area needed for the memory;
- (b) allocating a memory of the estimated size;
- (c) performing placement and routing; and
- (d) assessing whether the allocated memory is sufficient, and if not, incrementing the size of the memory and iterating again.

20 23 The computer readable medium of claim 22 wherein instruction (a) further includes the instruction of receiving a netlist comprising a logical description of cells and interconnects in a design as input.

24 The computer readable medium of claim 23 wherein instruction (a) further includes the instruction of generating the initial size estimate by counting a number of cells in the netlist and dividing by constant.

5 25 The computer readable medium of claim 24 wherein instruction (c) further includes the instruction of:

- (i) performing an initial placement of cells in the allocated memory;
- (ii) determining if the initial placement of cells in the allocated memory is successful;
- (iii) if the initial placement of cells in the allocated memory is successful, performing global routing for large signals;
- (iv) determining if the global routing in allocated memory is successful; and
- (v) if the global routing and allocated memory is successful, performing detail routing for local nets.

26 The computer readable medium of claim 25 further including the instruction of using a heuristic algorithm to determine success and failure.

27 The computer readable medium of claim 26 further including the instruction of incrementing the allocated memory by a minimum amount when failure is determined.

28 The computer readable medium of claim 27 further including the instruction of incrementing the allocated memory by one row at a time until it is determined that the allocated memory can be placed and routed without size or timing violations.

5

29 A method for automating an ASIC design flow, comprising the steps of:

- (a) providing a server over a network that integrates a set of design tools, including an automated front-end software process and an automated back-end software process;
- (b) allowing a user to access the server over the network and enter a request for an ASIC design;
- (c) executing the front-end software process to automatically generate a netlist for the design from the user request; and
- (d) executing the back-end software process to automatically generate a placement and route view of the ASIC.

10  
15  
20  
25  
30  
35  
40  
45  
50  
55  
60  
65  
70  
75  
80  
85  
90  
95  
100  
105  
110  
115  
120  
125  
130  
135  
140  
145  
150  
155  
160  
165  
170  
175  
180  
185  
190  
195  
200